A Global Approach to Kinematic Path Planning to Robots with Holonomic and Nonholonomic Constraints

Adam Divelbiss, Sanjeev Seereeram, John T. Wen
Rensselaer Polytechnic Institute
Troy, New York

A Global Approach to Kinematic Path Planning to Robots with Holonomic and Nonholonomic Constraints

Adam Divelbiss Sanjeev Seereeram John T. Wen
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
divelbis@cat.rpi.edu seereeram@ral.rpi.edu wen@ral.rpi.edu

Abstract

Robots in applications may be subject to holonomic or nonholonomic constraints. Examples of holonomic constraints include a manipulator constrained through the contact with the environment, e.g., inserting a part, turning a crank, etc., and multiple manipulators constrained through a common payload. Examples of nonholonomic constraints include no-slip constraints on mobile robot wheels, local normal rotation constraints for soft finger and rolling contacts in grasping, and conservation of angular momentum of in-orbit space robots. The above examples all involve equality constraints; in applications, there are usually additional inequality constraints such as robot joint limits, self collision and environment collision avoidance constraints, steering angle constraints in mobile robots, etc.

This paper addresses the problem of finding a kinematically feasible path that satisfies a given set of holonomic and nonholonomic constraints, of both equality and inequality types. The path planning problem is first posed as a finite time nonlinear control problem. This problem is subsequently transformed to a static root finding problem in an augmented space which can then be iteratively solved. The algorithm has shown promising results in planning feasible paths for redundant arms satisfying Cartesian path following and goal endpoint specifications, and mobile vehicles with multiple trailers. In contrast to local approaches, this algorithm is less prone to problems such as singularities and local minima.

1 Introduction

Controlling robot motion, including both manipulators and mobile vehicles, usually involves the following steps:

1. Kinematic path planning: find a path that satisfies all the geometric specifications and constraints of a given task.
2. Trajectory generation: index the path with time to generate a dynamic trajectory.
3. Dynamic trajectory following: design a servo controller (possibly incorporating dynamic information of the overall system) to follow the trajectory.

This paper focuses on the kinematic path planning problem. In contrast to most of the existing algorithms which are local and reactive in nature, our approach is a global one which warps an entire path to satisfy all the constraints. A common classification of constraints involves holonomic versus nonholonomic. If a constraint can be expressed in terms of the generalized coordinate, it is holonomic; if a constraint involves the generalized velocity and it is not integrable, then the constraint is nonholonomic. Examples of holonomic constraints include a manipulator constrained through its contact with the environment, e.g., inserting a part, turning a crank, etc., and multiple manipulators constrained through a common payload. Examples of nonholonomic constraints include no-slip constraints on mobile vehicle wheels, local normal rotation constraints for soft finger contacts in grasping, and conservation of angular momentum of space robots. There may be other design constraints imposed by the task, for example, equality constraints such as the desired terminal configuration, specified end effector path, etc., and inequality constraints such as manipulator joint limits, self collision and environment collision avoidance constraints.

There is abundant literature on path planning for redundant robots, which are examples of systems with holonomic constraints, and mobile robots which are examples of systems with nonholonomic constraints, but seldom on both. The principal reason is that in the local approach, the two problems are fundamentally different in that redundant robots can in general move in all directions locally in the configuration space, but nonholonomic systems can only move in certain directions. Consequently, the issues related to redundant robots are singularity, redundancy resolution, joint cyclicity for cyclic end effector paths, etc., and, for nonholonomic systems, the main issue is in finding a path whose tangent lies within the admissible directions. There are also commonalities in the two classes of problems:

1. Kinematic models are linear in control (i.e., admissible velocities).
2. Collision avoidance and joint limits are represented as a set of inequality constraints.

In our approach, the entire path is iterated toward a solution. Therefore, in this framework, whether the constraint is holonomic or nonholonomic does not make a fundamental difference; a more important distinction is between equality and inequality constraints. Indeed, path planning for redundant robots and mobile robots are treated in exactly the same way in our proposed approach.

The manipulator path planning problem is traditionally based on geometric approaches. (Comprehensive surveys can be found in [1, 2, 3].) The majority of these use the configuration representation of the manipulator and the obstacles and joint limits (as originally proposed in [4]). In this formulation, the path planning problem is reduced to finding a feasible path for a single point. Various tools from geometry, topology and algebra have been utilized to develop methods such as roadmap, cell decomposition, and potential field methods. The drawback of the configuration space approach is that the configuration space can become high dimensional for manipulators with several joints and mapping out the free region for a redundant arm in a cluttered environment is a time consuming process and produces large graphs to be searched. Recently, some work has emerged which focuses on minimizing this growth in computation by incorporating heuristics [5, 6].

The specific issue of redundancy resolution in redundant manipulators has traditionally been addressed by using the Jacobian pseudo-inverse [7, 8, 9, 10], which is a local, or point-wise on the path, approach. This approach is simple to use, has low computational overhead, and can be combined with the local potential field for collision avoidance. The drawbacks include local minima, breakdown near or at singularities, and non-cyclic or even unstable joint motion for cyclic end effector paths [11, 12, 13, 14]. Some global techniques based on optimization have also been proposed [15, 16, 17, 18, 19], but the resulting two-point boundary value problem (TPBVP) is difficult to solve for complex problems.

In the rapidly accumulating literature on nonholonomic motion planning, there appears to be a similar dichotomy of approaches: graph search based methods as in [20, 21, 22, 23, 24, 25, 26] and analytic methods [27, 28, 29, 30, 31, 32, 33, 34, 35]. The former class tends to be computationally inefficient but can handle general constraints, the latter class gives elegant insights into the structure of the solution but may generate impractical paths. Recently, some important insights have been gained by linearizing the kinematics equation about a non-stationary trajectory instead of a fixed equilibrium. It is shown in [36] that the linearized time varying system is frequently controllable, and a locally stable feedback controller can be designed. In [37], a globally stable feedback controller is also found. Based on this approach, a general procedure for designing globally stable time varying feedback controller has been obtained [38].

A common starting point for the analytic approaches to the path planning problem is to pose it as a general nonlinear control problem. Using this framework, we present a new method of solving the path planning problem in this paper. The formulation is based on converting the nonlinear control problem into a nonlinear algebraic equation. The problem then becomes a nonlinear root finding problem in which the dimension of the search space is very high (infinite if $\{u(t) : t \in [0,1]\}$ is taken from an infinite dimensional functional space) compared to the number of equality constraints (for the end point constraint). The nonlinear root finding problem is further converted to an initial value problem (IVP) which is much easier to solve than the usual TPBVP typically arising in the optimization approach. Under some additional assumptions, the IVP can be shown to be wellposed [39] and a variable step size ODE solver is used to propagate the solution. Examples ranging from a front-wheel driven car to triple trailers to nine degree-of-freedom (DOF) manipulators have been successfully tackled [40, 41, 42]. A similar approach has also been proposed independently in [43, 44] for kinematic path planning with only the end point constraint. For this case, the wellposedness property of the IVP is shown to be generic.

Since the dimension of the search space is very high, there are many possible solutions to the root finding problem which results from the path planning problem. For any practical applications, additional constraints must be placed. We have adopted an approach similar to the global exterior penalty function method [45, 46] which converts inequality constraints into a zero finding problem. This differs from the familiar artificial potential field method which is an interior penalty function (or barrier function) method in that the initial guess may be infeasible.

The inequality constrained case can then be combined with the equality constraints into an augmented zero finding problem. Non-configuration space constraints, for example, constraints involving corners of a vehicle. body of the robot, etc., can also be

incorporated in this formulation.

The global path planning approach that we have proposed has been applied to examples involving redundant manipulators and nonholonomic vehicles. We note the following attractive features of this algorithm:

1. Both equality and inequality constraints can be included in this formulation. The constraints can be nonlinear in the configuration variable, so task space constraints (which involve nonlinear kinematic function of the configuration variables) are also allowed.
2. The initial guess does not have to be feasible. The planner iteratively warps the path until all constraints are satisfied.
3. This approach emphasizes feasibility over optimality in contrast to other global approaches. Once a feasible solution is found, optimality can be incorporated as a secondary constraint.
4. The IVP formulation is computationally easier to solve than the TPBVP.
5. There are additional points related specifically to redundant manipulators:
   - Goal task variability: This approach can be used for finding a feasible joint sequence from a fixed initial configuration to a specified Cartesian or joint space goal – the ***path planning*** problem – as well as for global ***redundancy resolution*** along a specified Cartesian path.
   - Singularity robustness: The global nature of the planner avoids the Jacobian singularity problem inherent in local methods. While the controllability about a configuration is lost at the singularity, the algorithm can proceed as long as controllability about the planned path, which is a much looser condition to satisfy, is retained.
   - Incorporation of additional equality constraints: As an application of this capability, the planner can generate cyclic joint space motion for a specified cyclic task space motion.

   There are also additional points related specifically to nonholonomic systems:
   - The planner only requires local controllability about a path in every iteration but does not require controllability about a configuration. This is important since the latter is not satisfied for nonholonomic systems.

The rest of this paper is organized as follows: Section 2 describes the theory behind the proposed algorithm. Section 3 describes the global exterior penalty function method to handle inequality constraints. Section 4 shows a number of simulation examples involving path planning for redundant manipulators and mobile vehicles with trailers.

2 Kinematic Path Planning Subject to Equality Constraints

In this paper, we consider the problem of finding a continuous path that links the specified initial and final configurations while satisfying a set of specified equality and

inequality constraints (which may be either holonomic or nonholonomic). To state this problem precisely, we first represent a kinematic model as a control system:

$$\dot{q}(t) = u(t) \quad ; \quad q(0) = q_0 \tag{1}$$

where $q \in \mathbf{R}^n$ is the configuration variable and $u \in \mathbf{R}^n$ is a pseudo-velocity considered as the control input. Note that since this is a path planning problem, the path variable, t, is arbitrarily normalized to be within the interval $[0, 1]$. There may also be additional holonomic and nonholonomic constraints imposed along the path. If the constraints along the path are directly incorporated in the kinematics equation, these constraints are treated as ***hard constraints.*** If the planner iteratively updates the path until the constraints are within a certain tolerance, these constraints are considered as ***soft constraints.*** We first state the general path planning problem with soft constraints:

> *Define* $\underline{u} = \{u(t) : t \in [0,1]\}$ *and* $\underline{q} = \{q(t) : t \in [0,1]\}$. *Given (1), find* $\underline{u} \in L_2([0,1]; \mathbf{R}^n)$ *such that* $q(t)$ *satisfies (1) and* $\underline{c}(\underline{q}, \underline{u}) = 0$ *where* $\underline{c} : L_2([0,1]; \mathbf{R}^n) \times L_2([0,1]; \mathbf{R}^n) \rightarrow \mathbf{Y}$ *is a given equality constraint function for a specified normed linear space,* $\mathbf{Y}$.

The constraint function $\underline{c}$ may include physical constraints, such as nonintegrable velocity constraints on wheels (nonholonomic) and contact constraints of manipulators (holonomic), or artificial constraint such as the desired end effector path of a redundant manipulator. Pathwise holonomic constraints can be represented as

$$\underline{c}(\underline{q}, \underline{u})(t) = k_1(t, q(t)) \qquad , t \in [0,1] \tag{2}$$

and pathwise nonholonomic constraints are of the form

$$\underline{c}(\underline{q}, \underline{u})(t) = k_2(t, q(t), \dot{q}(t)) \qquad , t \in [0,1] \tag{3}$$

and can not be integrated to a constraint only involving the configuration variable. In many applications, k_2 is linear in $\dot{q}$ and invariant in t, i.e., $k_2(t, q(t), \dot{q}(t)) = K_2(q(t))\dot{q}(t)$. A system may also be subject to what is known as second order nonholonomic constraints (for example, for a manipulator with unactuated joints [47]).

For the path planning problem with hard constraints, the path constraints in (2) and (3) are explicitly removed. In the holonomic case, write (2) in the differential form:

$$\frac{\partial k_1(t,q)}{\partial t} + \frac{\partial k_1(t,q)}{\partial q} u = 0. \tag{4}$$

Denote the Jacobian matrix $\frac{\partial k_1(t,q(t))}{\partial q}$ by $J(t,q)$. If J is a fat matrix (as in the case of redundant manipulators) and is nonsingular, then the constraint can be explicitly incorporated in the kinematics, resulting in

$$\dot{q} = -J^+(t,q)\frac{\partial k_1(t,q)}{\partial t} + \tilde{J}(t,q)v \tag{5}$$

where $J^+(t,q)$ denotes the Moore–Penrose pseudo–inverse of $J(t,q)$, $\tilde{J}(t,q)$ is an arbitrary full rank matrix whose range coincides with the null space of $J(t,q)$, and v is the new effective control variable.

In the nonholonomic case, if the constraint is linear in u, then the constraint can be eliminated, resulting in a new kinematic equation:

$$\dot{q} = \widetilde{K}_2(q)v \tag{6}$$

where $v \in \mathbf{R}^m$ is the new effective control variable (m is the dimension of the null space of $K_2(q)$, assuming it is a constant).

The general planning problem can now be restated as before except the kinematic equation is modified as below to incorporate the path constraints:

$$\dot{q}(t) = h(t, q(t)) + f(t, q(t))u(t) \quad ; \quad q(0) = q_0. \tag{7}$$

The drift term, $h(t, q)$, is zero in the case of nonholonomic constraints linear in $\dot{q}$. The equality constraint function may involve only the end configuration:

$$\underline{c}(\underline{q}, \underline{u}) = q(1) - q_d \tag{8}$$

where q_d is the desired final configuration. In the case of redundant manipulators, joint cyclicity implies $q_d = q_0$.

For the path planning problem as formulated above with either soft or hard constraints, our approach is based on converting the differential description of the problem into an algebraic form. First consider the soft constraint formulation. Recall this case involves a very simple kinematic equation given by (1). For a given initial configuration q_0, $\underline{q}$ can be related to $\underline{u}$ via a linear causal map, $\underline{D}$, and an initial condition vector, $\underline{q}_0$:

$$\underline{q} = \underline{D}\,\underline{u} + \underline{q}_0. \tag{9}$$

Define the constraint error as

$$\underline{y} = \underline{c}(\underline{D}\,\underline{u} + \underline{q}_0, \underline{u}). \tag{10}$$

The path planning problem is now reformulated as a nonlinear root finding problem for $\underline{c}$ as a function of $\underline{u}$ Nonlinear root finding has the reputation being numerically challenging [48, §9.6]. However, the situation here is different than the general case in that the dimension of the search variable, $\underline{u}$, is typically much larger than the dimension of the constraint equation. Consequently, there are a very large number of roots and finding one of these roots is less difficult than the general root finding problem. However, equality constraints alone may not produce physically realizable solution; we shall see how this formulation can be extended to inequality constraints in the next section.

To find a $\underline{u}$ that solves $\underline{y} = 0$, our basic strategy is to lift a path in $\mathbf{Y}$ that connects an initial guess $y(0)$ to zero to a path in $L_2([0,1]; \mathbf{R}^m)$. Then the end point of the path in $L_2([0,1]; \mathbf{R}^m)$ is a solution that satisfies the stated equality constraint $c(\underline{q}, \underline{u}) = 0$. To achieve this, we set up the path iteration equation:

$$\frac{d\underline{y}}{d\tau} = G\frac{d\underline{u}}{d\tau} \tag{11}$$

where

$$G = \nabla_{\underline{q}}\underline{c}\ \underline{D} + \nabla_{\underline{u}}\underline{c}.$$

If G is full rank, or, equivalently, the null space of G^* (the adjoint map of G) is zero, then any one of the following algorithms can be applied to update $\underline{u}$:

$$\frac{d\underline{u}}{d\tau} = G^+ \frac{d\underline{y}_d}{d\tau} + \tilde{G}\xi \tag{12}$$

where G^+ is the Moore–Penrose pseudo–inverse of G, and $\underline{y}_d$ is any desired convergence profile of $\underline{y}$ in τ and $\underline{y}_d(0) = \underline{y}(0)$, for example, we can choose $\underline{y}_d(\tau) = e^{-\alpha\tau}\underline{y}(0)$, for exponential convergence. However, this rate may not correspond to the physical convergence rate since (12) may take much longer to solve. In practice, Eq. (12) may be solved by an ODE solver or discretized with the time step in τ found by line search.

In the hard constraint formulation and with only the end configuration constraint (i.e., $\underline{c}(\underline{q}, \underline{u}) = q(1) - q_d$), the equality constraint error can be written as

$$\underline{y} = F(\underline{u}) - q_d \tag{13}$$

where F maps the control, $\underline{u}$, (for a given q_0) to the final configuration. The analytic form of F is in general difficult to find, and will not be explicitly required. The system (7) is globally controllable if and only if the nonlinear map F is onto, and the system is locally controllable around $\underline{u}$ if and only if $\nabla_{\underline{u}} F(\underline{u})$ is a linear onto map. Local controllability around $\underline{u}$ is equivalent to the controllability of the linear time varying system obtained after linearizing (7) around $\underline{q}$ and $\underline{u}$. The path planning problem can be iteratively solved as before using (12) but with

$$G = \nabla_{\underline{u}} F(\underline{u}).$$

A sufficient condition for convergence is that G is full rank for all τ, or, equivalently, the time varying linearized system with $\underline{u}(\tau)$ is controllable. For systems without drift, such as nonholonomic systems, it has been shown in [43] that this condition is generic.

Equation (12) can be solved by an ordinary differential equation (ODE) solver with an initial guess of $\underline{u}(0)$ such that the full rank condition is satisfied. In addition, $\underline{y}_d(\tau)$ can be used to check the accuracy of the solution, $\underline{y}(\tau)$, at each τ. It can be shown that if $\underline{u}(0)$ is sufficiently close to a solution, then the wellposedness of the IVP (and its convergence) is assured. As stated before, the wellposedness of the IVP in the hard nonholonomic constraint case has been shown to be generic and there has been a surge of recent interest of this technical issue [39]. For the general problem, the wellposedness condition remains an important research topic. In all of our simulation experience involving equality constraints, this has never been a problem. In the next section, when constraints are included through the global exterior penalty function, the issue of wellposedness becomes more severe.

An interesting aspect of this approach is that in (12), ξ does not affect the guaranteed convergence rate (specified by α), though it does affect the way $\underline{u}$ converges. Since the dimension of $\underline{u}$ is much larger than $\underline{y}$, there is much freedom in ξ to affect the eventual convergent solution. For example, ξ may be chosen so that additional constraints in $\underline{u}$ may be satisfied. We have explored choosing ξ via quadratic programming with some success, but more research needs to be done in exploiting this degree of freedom.

There are other possible choices of $\underline{u}$, for example, in the soft constraint case,

$$\frac{d\underline{u}}{d\tau} = G^* \frac{d\underline{y}_d}{d\tau} + \tilde{G}\xi \tag{14}$$

$$\frac{d\underline{u}}{d\tau} = \frac{G^* \underline{y}}{\left\| G^* \underline{y} \right\|^2} \frac{dw_d}{d\tau} + \tilde{G}\xi \qquad w_d = \frac{1}{2} \left\| y_d \right\|^2 . \tag{15}$$

The trade–offs between computational load and convergence speeds between these schemes are yet to be fully explored.

The initial guess $\underline{u}(0)$ will clearly affect the convergence. We have not extensively explored an intelligent procedure (perhaps based on past experience) for this selection. In all of our examples, the initial conditions are simply chosen to avoid the rank deficiency of G.

In the algorithm described above, it is critical to be able to calculate G efficiently. Since G relates an infinitesimal variation $\delta\underline{u}$ to a corresponding infinitesimal variation $\delta\underline{c}$, it can be computed based on the linearized kinematic equation (and the constraint equation in the soft constraint case). This procedure is described in detail in [49], and has been used in all of our examples.

3 Kinematic Path Planning Subject to Inequality Constraints

In the previous section, only equality constraints are considered. However, for realistic problems, there are also many inequality constraints which need to be enforced. For problems involving equality constraints only, the obtained path is frequently undesirable. For example, in the case of a tractor with twin–trailers, the front wheel angles of the tractor may go through an unrealistically large range motion, the trailers can become jack–knifed, and the entire vehicle may go through several complete revolutions; in the redundant arm case, the joints can violate joint limits, and there may be self collision or collision with other objects in the workspace. Clearly, if this algorithm is to be used in practical applications, additional constraints must be incorporated. In this section, we present an approach similar to the exterior penalty function method of [45] and which has so far been shown to be very effective in addressing the inequality constraint issue.

There are three basic approaches to address the inequality constraints:

- Convert the inequality constraints into equality constraints by defining a function that is zero when the constraint is satisfied and non–zero when the constraint is not satisfied. Once this is done, the zero finding approach of the previous section can be applied to find a zero solution which corresponds to a feasible path.
- If the feasible region is convex polyhedral, then linear programming or quadratic programming can be used to select the free variable ξ in (12) without affecting the convergence of the equality constraints.
- Pre–warp the vector with a multiplicative potential function $\phi(q)$ which is zero on the boundary of the inequality constraint and positive inside the feasible region. It has recently been pointed out in [44] that as long as the initial path $\underline{q}(0)$ lies strictly within the feasible region, then the same algorithm (12) can be applied to iteratively move $\underline{y}$ to zero.

We have extensive experience with the first two approaches. The first approach has proven to be very effective, though there are currently no (genericity) results on the

full rank gradient condition as in the equality constraint case. The second approach is computationally intensive due to the need to repeatedly solve a constrained optimization problem. It is perhaps best used as a local optimization refinement in conjunction with the first approach. The third approach has just recently been suggested and is computationally untried; we intend to pursue this direction in the proposed research. For the remainder of this section, we will elaborate on the exterior penalty function approach.

Suppose the feasible region is defined by a set of p inequalities:

$$g_1(\underline{q}, \underline{u}) \leq 0 \tag{16}$$

where g_1 is nonlinear and assumed to be as smooth as needed, and $\leq$ is interpreted as a component-wise relationship. These constraints may be directly placed on the configuration variables such as joint limits or on other variables such as the end effector of a manipulator, vehicle boundaries, etc. Without loss of generality, we shall consider the hard constraint case for the discussion below. Let the relationship between $\underline{q}$ and $\underline{u}$ be denoted by $\underline{q} = \underline{F}(\underline{u})$. By substituting this relationship, the constraint inequalities are transformed to

$$g(\underline{F}(\underline{u}), \underline{u}) \leq 0 \tag{17}$$

or more simply expressed as

$$g(\underline{u}) \leq 0. \tag{18}$$

Suppose now we wish to constrain the states of the system to stay within the feasible region defined by the task space. We then define a penalty function corresponding to the i^{th} system state as:

$$z_i = \sum_{j=1}^{N} \begin{cases} \gamma_{ji} g_{ji}^{\lambda}(\underline{u}) & g_{ji}(\underline{u}) > 0 \\ 0 & g_{ji}(\underline{u}) \leq 0 \end{cases} \tag{19}$$

where $\lambda > 1$, g_{ji}'s correspond to the constraint applied to state i at discrete time j and γ_{ji}'s are constant weights. This function is nonzero when state i violates the constraints at any time along the path, and is identically zero only when all constraints are satisfied. For this penalty function, the penalty imposed depends upon the constraint violation raised to the λ power. Therefore, the penalty function approaches infinity as the system makes ever increasing incursion out of the feasible region. Of course many other choices for the constraint function are also possible. For example, another penalty function which we use frequently is the following:

$$z_i = \sum_{j=1}^{N} \begin{cases} \gamma_{ji}(1 - e^{-\lambda g_{ji}(\underline{u})})^r & g_{ji}(\underline{u}) > 0 \\ 0 & g_{ji}(\underline{u}) \leq 0 \end{cases} \tag{20}$$

where $r, > 1$ and $\lambda > 0$. This penalty function is bounded by the sum of the γ_{ji}'s, and is therefore in some sense less harsh than the previous penalty function. In practice we have seen that the second penalty function gives faster convergence than the first, in cases where the vehicle makes large excursions outside of the feasible region. This advantage in convergence stems from the fact that $\nabla_{\underline{u}} z_i(\underline{u})$ is smaller for the second constraint function than for the first when large constraint violations occur, causing

the IVP to be more wellposed for the second constraint function. For either penalty function, the composite constraint vector is defined as $z = \left[\begin{array}{ccc} z_1 & \ldots & z_n \end{array} \right]^T$.

We now have a situation similar to the unconstrained case: ***Find*** $\underline{u}$ ***so that*** $\underline{y} = 0$ ***and*** $z = 0$. The same approaches described in Section 2 can now be applied. Using the hard constraint formulation, the new differential equation in $\underline{u}$ now becomes:

$$\frac{d\underline{u}(\tau)}{d\tau} = G_1^+(\underline{u}(\tau)) \left[\begin{array}{c} \underline{y}_d \\ z_d \end{array} \right] + \widetilde{G_1}(\underline{u}(\tau))\xi(\tau) \tag{21}$$

where,

$$G_1(\underline{u}) \triangleq \left[\begin{array}{c} G \\ \nabla_{\underline{u}} z(\underline{u}) \end{array} \right] \tag{22}$$

and $\underline{y}_d$ and z_d describe the desired path in τ that links the initial error to zero. The gradient of $z(\underline{u})$ can be easily obtained via chain rule and the linearized kinematic equation [49]. The convergence of this algorithm now requires the full rank condition on G_1 in each iteration. Note that in (21), there is again a free parameter $\xi(\tau)$. The extra degrees of freedom offered by $\xi(\tau)$ may be used to satisfy additional optimality considerations.

The penalty function formulation we have presented allows for a wide range of constraint types. Up to this point we have considered constraints which apply directly to the configuration variable of the system, but the formulation allows constraints to be applied to non–configuration variables as well. In any practical application it is not enough just to constrain the origin of the system inside a feasible region. The boundaries of the system must stay inside as well. For instance, for a single planar body, we relate any point p on the boundary to the configuration variable q by the nonlinear transformation:

$$\left[\begin{array}{c} p_x \\ p_y \end{array} \right] = \left[\begin{array}{c} q_1 \\ q_2 \end{array} \right] + \left[\begin{array}{cc} \cos(q_4) & -\sin(q_4) \\ \sin(q_4) & \cos(q_4) \end{array} \right] \left[\begin{array}{c} r_x \\ r_y \end{array} \right] \tag{23}$$

where r_x and r_y are the x and y positions of the boundary point in the body frame. The same penalty function which applied to q before now applies to p, $g_j(p(q_j(\underline{u})))$. A similar transformation can be obtained to relate boundary points of multi–bodied systems to the configuration variable.

The penalty function formulation also allows for a wide range of feasible region types. For instance, when the feasible region is polyhedral, g takes on particularly simple form:

$$g(\underline{q}) = A\underline{q} - b.$$

We have looked at this type of constraint extensively. In fact, this constraint type is always used when limiting wheel or jackknife angles on wheeled vehicles. This type of constraint is simple to use but the feasible region is necessarily convex.

The penalty function can be used to enforce both convex and non–convex, non–polyhedral constraints. For instance, supposed it is desired to drive the system while keeping boundary points outside of an circular region. The constraint g at each time j can then be expressed as:

$$g_j(p) = -(p_x - x_0)^2 - (p_y - y_0)^2 + r^2$$

where x_0 and y_0 are constants, r is the radius, and p is a boundary point as described in (23). We have had much success in using this formulation with cars and tractor-trailer vehicles and present an example in the next section.

In the preceding discussion, system constraints are expressed as analytic functions of the configuration variable. But again, in practical situations it may not be possible to adequately represent the constraints by analytic functions. In particular, for highly unstructured environments it may require great effort to find an appropriate analytic function, and once found, it may be computationally intensive to apply it. Therefore, we propose a method based upon the contour map built up from the task space obstacle boundaries. First, suppose that the feasible region is defined by a set of p inequalities of the form:

$$g_2(\underline{F}(\underline{u})) \leq 0 \tag{24}$$

where g_2 is nonlinear, smooth as needed, and $\leq$ is component-wise. The gradient of g_2 with respect to $\underline{u}$ is then obtained by applying the chain rule,

$$\nabla_{\underline{u}} g_2 = [\nabla_{\underline{u}} \underline{F}]^T \nabla_{\underline{q}} g_2. \tag{25}$$

Note that in this equation, $\nabla_{\underline{u}} \underline{F}$ is independent of the constraints and that $\nabla_{\underline{q}} g_2$ depends solely upon the constraints. Therefore, rather than compute g_2 and $\nabla_{\underline{q}} g_2$ explicitly using analytic functions, it is possible to use a lookup table to compute these values. For g_2, the lookup table contains the cost for each point in a grid covering the task space. For $\nabla_{\underline{q}} g_2$ the table contains the gradient components for each point on the grid. As before, this formulation will also work for non-configuration variables. Since the exterior penalty function approach is used, the contour map is zero everywhere inside the feasible region and non-zero outside. We have just begun using this method for path planning and have so far had good success. An example of the contour map method applied to a car is given in the next section.

In this, and the previous sections, we have used the notation $\underline{u}$ to denote the set of control inputs applied to the system at each point in time along the path. For the continuous case this set contains an infinite number of elements. Therefore $\underline{u}$ in vector form would be an infinite dimensional vector. However, in order to implement the algorithm on a digital computer $\underline{u}$ needs to have finite length. The first obvious step would be to discretize $u(t)$ using the standard basis. We have used this approach extensively and have obtained good results from it. Another approach we have tried is to use the first N elements of the Fourier basis to approximate $u(t)$. In this formulation, for each discrete time j we represent the control input as:

$$u_j = \Phi(j\Delta t)\underline{\lambda} \tag{26}$$

where Φ is a matrix containing the Fourier basis elements and $\underline{\lambda}$ is the constant vector of Fourier coefficients. In this formulation, $\underline{\lambda}$ uniquely describes the control over all time, since it is independent of time. Now for a given initial configuration $\underline{q}_0$, $\underline{q}$ can be related to $\underline{\lambda}$ by a causal map, $\underline{F}_1$:

$$\underline{q} = \underline{F}_1(\underline{\lambda}). \tag{27}$$

Furthermore, substituting $\underline{F}_1$ for $\underline{F}$ and $\underline{\lambda}$ for $\underline{u}$ in all previous equations, we see that all previous methods still work. The full advantages and disadvantages in using

the Fourier basis formulation rather than the discrete basis is unknown at present. However, in the case of driving a car around a circle, we have observed that the discrete basis formulation has great difficulty arriving at a solution whereas the Fourier basis formulation solves the problem with relative ease.

4 Examples

We have applied the algorithm to a large number of computer simulation examples and experimentally to a mobile robot. For illustration purposes, we include several examples involving three different wheeled vehicles and a point–to–point path planning of a 4R planar arm and a 9DOF arm like the one in our lab (see next section on a fuller description of this arm).

In the first example, we consider the parallel parking of a double tractor–trailer. A true double tractor–trailer, like those seen driving on the highways, actually consists of a tractor with three trailers: two long trailers connected by a comparatively short trailer. The short trailer, or dolly, makes any backing–up situation extremely difficult in that small backward motions can produce large jackknife angles between the dolly and the trailers. In fact, it is known that with a human operator one must, in general, disconnect the end trailer and dolly before any backward motions are attempted.

In the example below, shown in Fig. 1–2 below, the front wheels are constrained to $\pm 35°$ and the jackknife angles (the jackknife angle is defined as the angle between the center line of one trailer and the center line of an adjacent trailer) are constrained to $\pm 50°$. The limits on the jackknife angles are sufficient to ensure that none of the trailers collide. In this example the algorithm required about four hundred fifty evaluations of the right hand side of the differential equation to achieve a tolerance on the potential fields of 0.01. The penalty function method used here is simply the first convex, polyhedral method mentioned in the previous section.

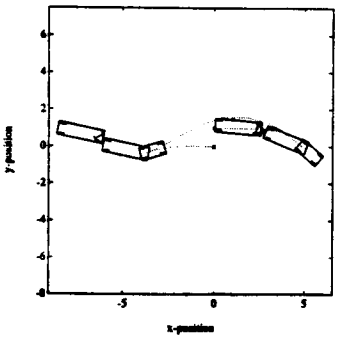

Figure 1: Parallel Parking Path of a Double Tractor–Trailer with Constraints

The next example is presented to demonstrate the versatility of the algorithm with regard to the type of constraint applied. In this example, it is desired to drive a tractor–trailer vehicle around a circular obstacle while remaining within a rectangular region limiting motion along the x–axis. To further complicate the example, two other

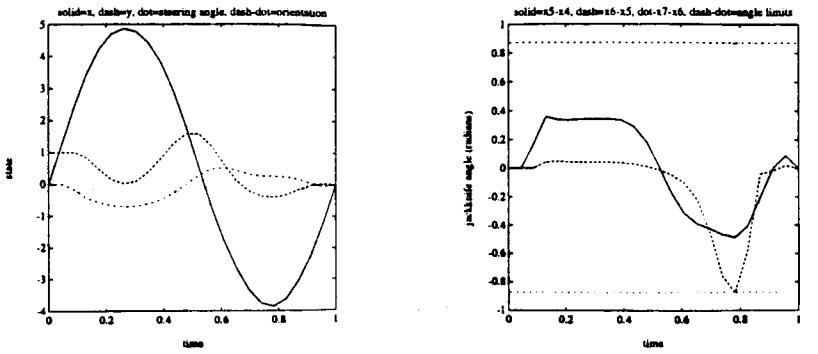

Figure 2: Parallel Parking Path of a Double Tractor–Trailer with Constraints

circular obstacles are included above and below the first one. In addition, these task space constraints are applied to fourteen different non–configuration variable points around both the boundary of the tractor and the boundary of the trailer. The steering angle is constrained to $\pm 20°$, and the jackknife angle is constrained to $\pm 50°$. We therefore have in this one example, all of the different types of constraints mentioned in the previous section: convex polyhedral (steering, jackknife, and x-position), non–convex non–polyhedral (circular obstacles), and non–configuration variable (boundary point) constraints. The Fourier basis representation of the control input, with thirty three basis elements, was also used in this example. Using a steepest descent method with a Golden section search, the algorithm took fifty four evaluations of the right hand side to converge. The initial path in this case, drives straight through the obstacle. The final result is as shown below in Fig. 3.

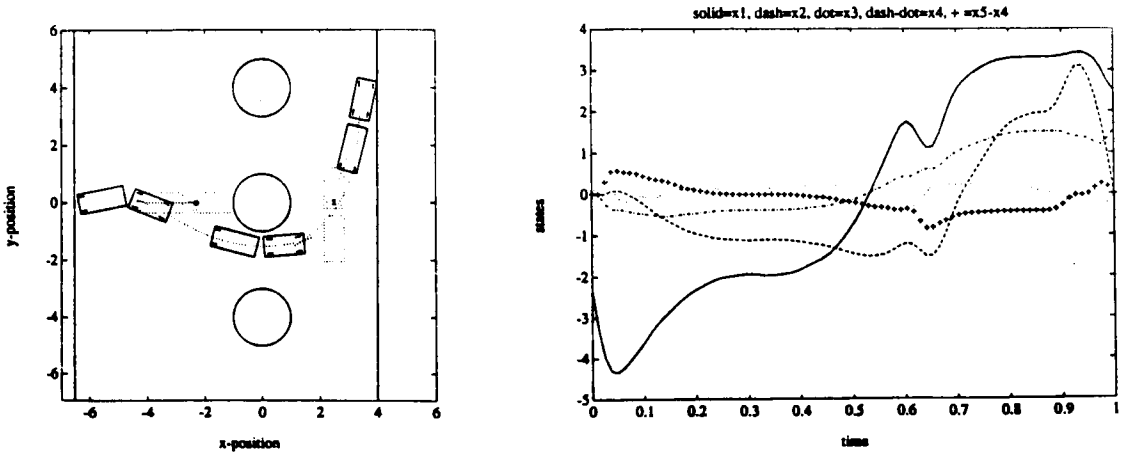

Figure 3: Tractor–Trailer Example with Non–Convex Task Space Constraints

The next simulation example uses the contour map method to calculate the penalty

function. In this example, it is desired to drive a car around a circular obstacle while remaining inside of a circular region. Seven points around the boundary of the car are used for non-configuration variable constraints. The front wheels are constrained to $\pm 15°$. The Fourier basis was once again used to represent the control input along the path. This example took about one hundred evaluations of the right hand side using a discretized version of (21) with Golden section search to find the optimal step size. The initial path drives straight through the obstacle. Below, the first set of plots, in Fig. 4, show the car and path in the context of the task space. The second set of plots, in Fig. 5 show the contour map used in this example.

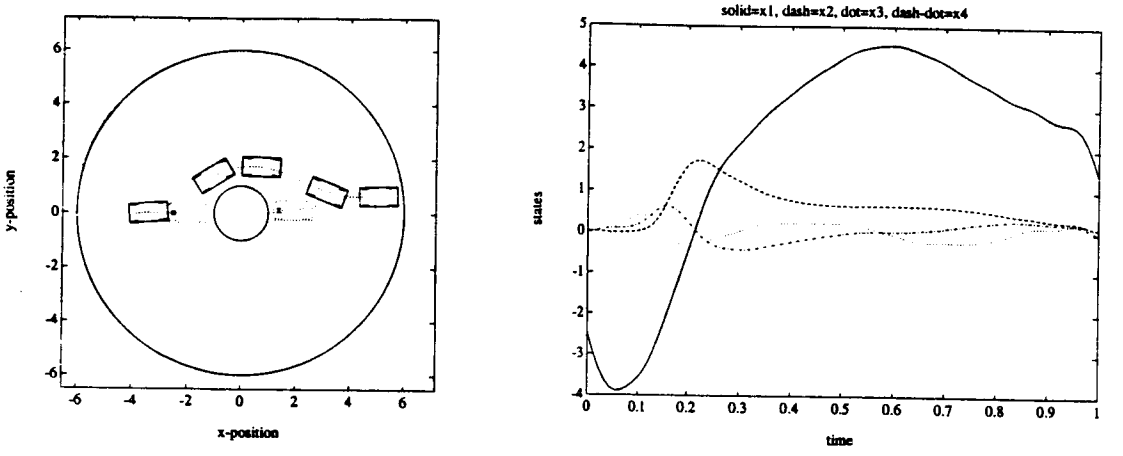

Figure 4: Path Planning with Contour Map Method

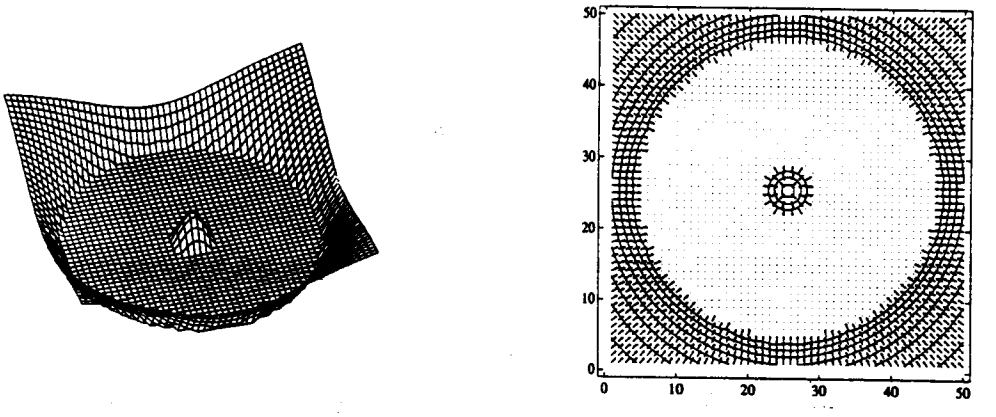

Figure 5: Plots of Exterior Penalty Function and Contour Map

In the final wheeled vehicle example, we present experimental results of applying the

path planning algorithm to an actual mobile robot. The robot is actually a one quarter scale model car, complete with a passive suspension system, on–board IBM compatible 486 33–MHz computer, and wheel encoders for use in dead–reckoning feedback. The desired path was a simple parallel parking path with constraints only on the steering angle of $\pm 15°$ which is the physical limit of the car. The purpose of this experiment was to determine if a real robot can actually follow a path generated by the planning algorithm. In this example, the path was generated, and then a velocity profile was imposed. A simple, Lyapunov function based, feedback controller was used to try and track the path. Although there is some error in tracking the path, it is obvious from the plots below, in Fig. 6–7 that the path is such that the real car can follow it. The tracking errors are due in part to error in the dead–reckoning feedback and to the controller used.

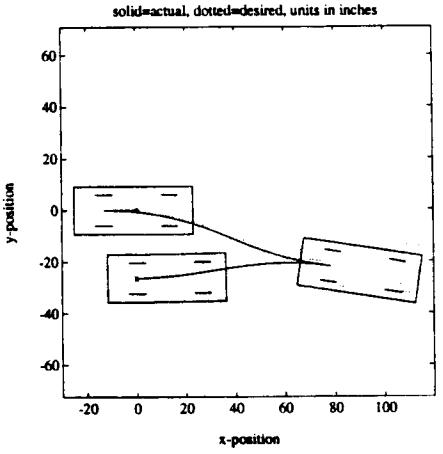

Figure 6: Parallel Parking of Catmobile: Experimental vs. Simulation Path

Various examples of redundant manipulators are presented which incorporate both joint and task space inequality constraints. Key features of this approach include various possible goal task specifications (from end-point only to entire path specification), joint path cyclicity, and robustness to manipulator singularities. Inequality constraints are handled by the global penalty functions and a linear inequality set description as shown in Sections 3. Simulations were performed using Matlab with computationally intensive routines coded as Matlab-callable C routines (cmex files). Note that the end effector constraint is treated as a soft constraint here as compared to the nonholonomic constraint considered above which is treated as a hard constraint.

The first example illustrates the joint sequence for a 3R planar arm required to trace out the tip path shown while remaining within a set of task space boundaries (Fig. 8). The path shown in Fig. 8 includes an intermediate joint vector in which links 2 and 3 become aligned – equivalent to a pose switch for spatial arms. Local planning methods typically encounter difficulty in handling pose changes since they correspond to arm singularities, and the arm Jacobian losing rank. The present algorithm executes the pose switch smoothly, while tracking the desired tip path. In the next situation (Fig. 9) a 4R arm is required to traverse to specified coordinates X_f. The obstacles are chosen

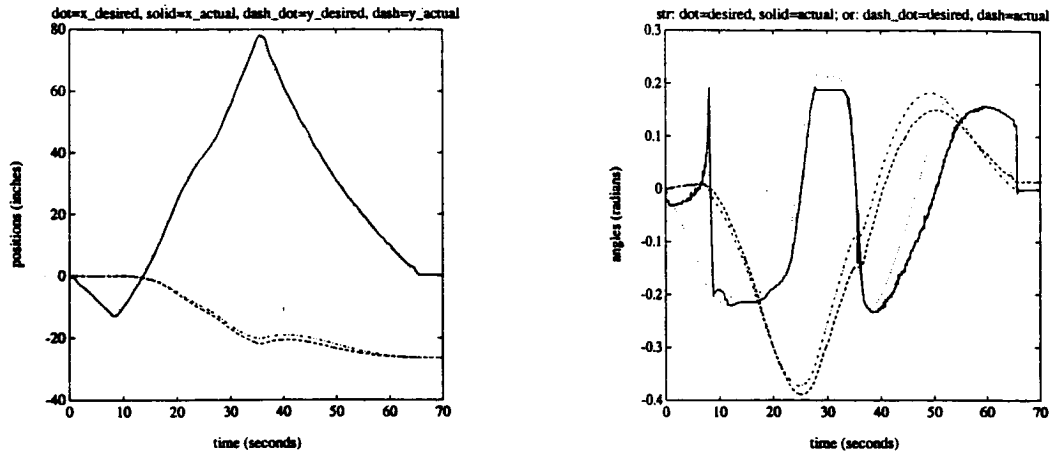

Figure 7: Parallel Parking of Catmobile: Experimental vs. Simulation State Trajectories

to provide a relatively narrow opening through which the arm must pass. Typically, this type of scenario is very challenging for planners based on purely local potential field formulations due to a local minimum formed in the space between Obstacles 1 and 2. By iterating from an intermediate path sequence generated for the goal endpoint without consideration of the obstacles, it is possible to warp the intermediate path to meet all the constraints. It should also be noted that the final path cannot be accomplished without switching the pose along the way.

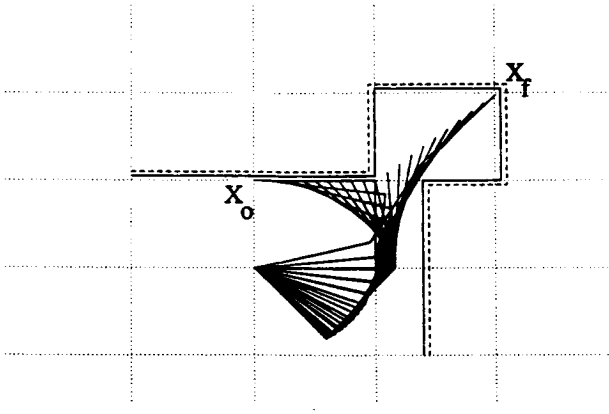

Figure 8: Cartesian Tip Path

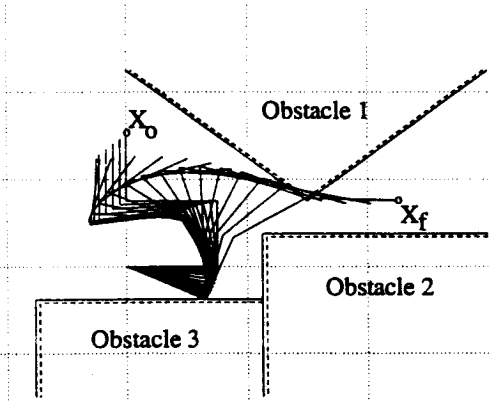

Figure 9: Obstacle Avoidance Problem

The ability to incorporate joint path cyclicity as an equality constraint is illustrated by an example where a 3R arm must trace out the boundaries of a Cartesian square path, within a restricted workspace (Fig. 10). Path planning for cooperating arms manipulating a common payload imposes an additional kinematic closure constraint on the arm tips. Fig. 11 shows the path sequence generated for a pair of 3R planar manipulators required to move the connecting linkage from its initial position of $(-1, 2, 0°)$ to

a goal position of $(2, 2, 45°)$. In the planar examples shown, the apparent collisions of the links with the obstacle boundaries arise out of the fact that checking is performed for the tips of each link only. A more complete collision detection procedure is being developed for spatial arm planning.

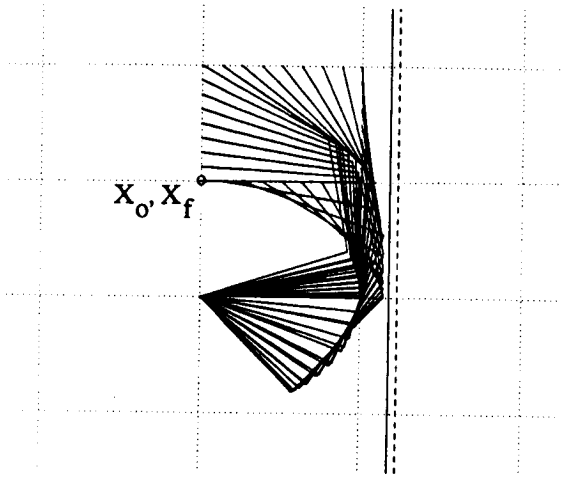

Figure 10: 3R Cyclic Path

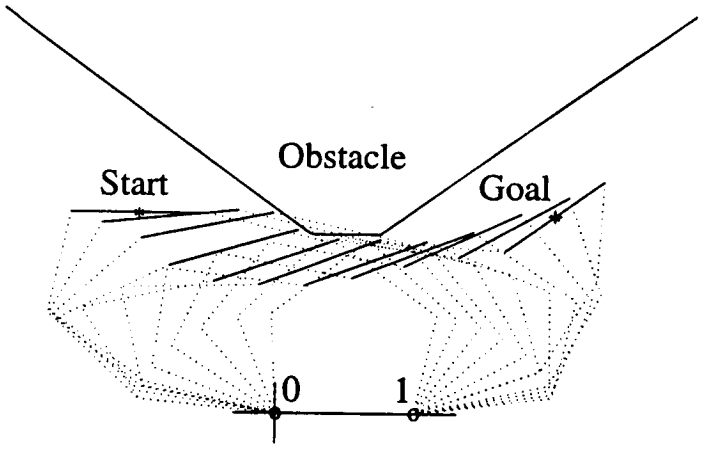

Figure 11: Cooperating Arm Problem

The remaining examples illustrate applications to a spatial redundant 9 DOF arm. This consists of a PUMA 600 manipulator mounted on a platform providing additional linear, rotate and tilt joints. (See [50] for details of the CIRSSE dual-arm robotic testbed.) Fig. 12 shows the output sequence for joint path end-point planning to a specified task space position/orientation in the presence of an obstacle. Fig. 13 shows the path sequence generated for a path following task incorporating a straight line translation coupled with a rotation of 180° about the tip Y axis. Because of the manipulator joint limits, this can only be accomplished by switching the PUMA's pose from ***elbow-up*** to ***elbow-down*** at some intermediate point. As in the planar examples, the present algorithm accomplishes a smooth transition between these arm configurations. The final example (Fig. 14) illustrates a cyclic joint path sequence computed for the arm to follow a circular task space path while maintaining a fixed tool orientation (perpendicular to the plane of the circle).

For the cases shown, the planar examples required from seven to ten iterations to reduce the task error and meet all the constraints, while the spatial examples ranged from ten to twenty-five iterations. Discretization levels ranged from $N = 10$ to 40 in size. For the 9 DOF redundant arm planning scenario, this represents a small computational load for the Sun SPARC-station used. workstations. For problems of larger size, this technique can be applied recursively to the desired discretization level, thereby keeping the individual iteration array operations small. The ability to apply the algorithm to the global problem (even at the coarsest resolution) is essential for

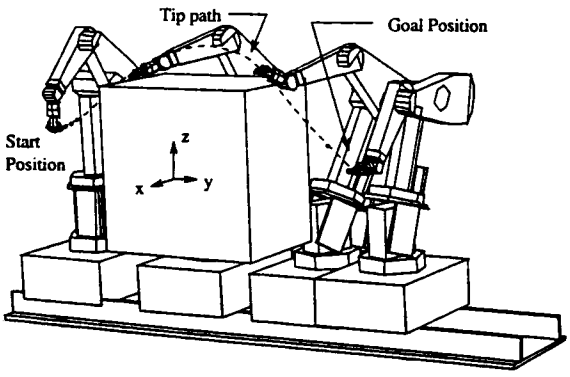

Figure 12: 9 DOF Path Planning

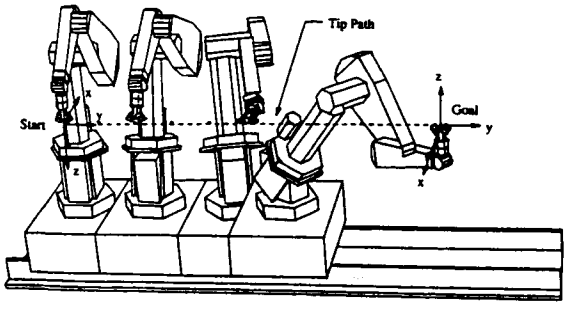

Figure 13: 9 DOF Path Following

developing globally feasible solutions to the constrained path planning problem.

5 Conclusions

Kinematic path planning for robots is a key step in their effective utilization on earth and in space. The various types of constraints such as holonomic and nonholonomic, and equality and inequality constraints pose particular challenges. In this paper, we consider a novel and promising method which warps the entire path until all constraints are satisfied. The main approach is to convert the differential (local) kinematic relationship to an algebraic (global) equation. With emphasis placed on feasibility rather than optimality, we obtain an initial value problem rather than two–point boundary value problem typically arisen in a global optimization approach. This formulation is general enough to include both redundant manipulators and nonholonomic systems, and combination of the two; these topics are traditionally treated separately due to their unique properties when local algorithms are applied. The inequality constraints are handled through a global exterior penalty function method, which allows for non–polyhedral constraints as well as constraints on non–configuration variables. For the future research, we will address the following fundamental issues related to this promising algorithm:

1. Develop conditions of convergence of the algorithm based on the wellposedness of the initial value problem. Also develop strategy to proceed when the algorithm fails to converge due to the rank deficiency of G in (12).
2. Develop an algorithm to adaptively adjust the path discretization step size based on the local discretization error.
3. Incorporate optimality as a secondary criterion.
4. Improve the sensitivity and robustness of the algorithm with respect to imperfection in the kinematic model.
5. Implement the planner on various experimental platforms.

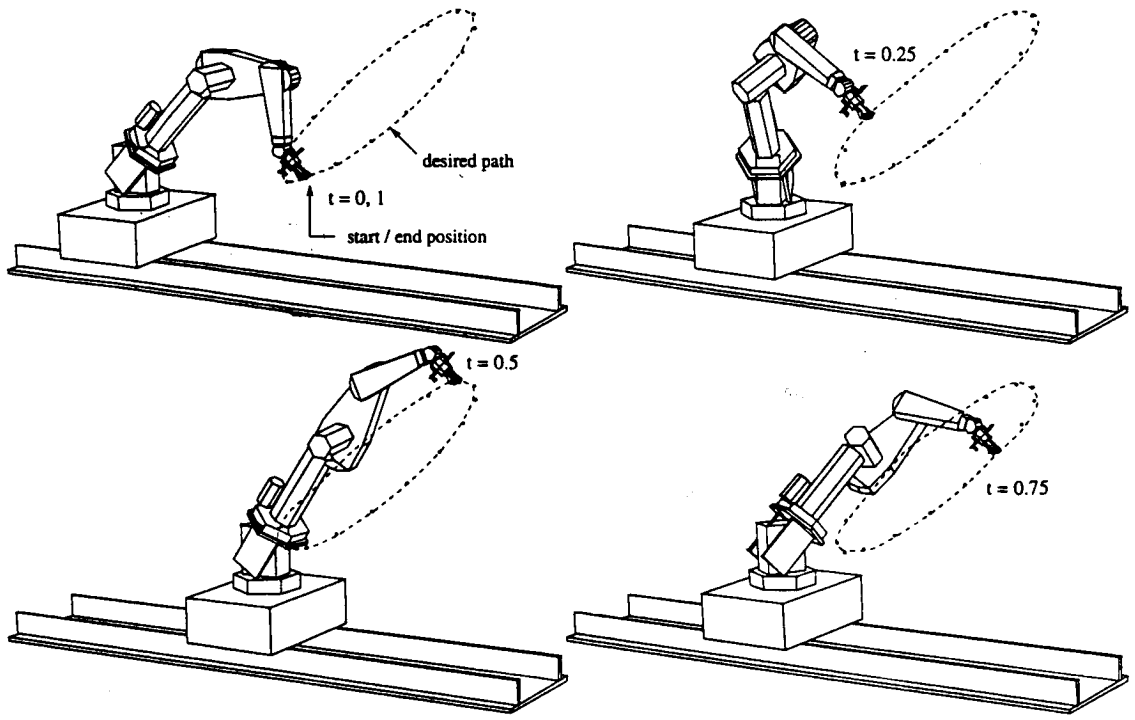

Figure 14: Spatial 9 DOF Cyclic Cartesian Path

Acknowledgment

This work is supported in part by the Center for Intelligent Robotic Systems for Space Exploration and New York State Center for Advanced Technology for Automation and Robotics at Rensselaer Polytechnic Institute.

References

[1] Lozano-Pérez T. and Taylor R.H. Geometric issues in planning robot tasks. In ***Problems of Robotics***. MIT Press, Cambridge, Mass., 1989.

[2] J.C. Latombe. ***Robot Motion Planning***. Kluwer International Series in Engineering and Computer Science: Robotics: Vision, Manipulation and Sensors. Kluwer Academic Publishers, 1991.

[3] Hwang Y.K. and Ahuja N. Gross motion planning – a survey. ***ACM Computing Surveys***, 24(3):219–291, 1992.

[4] Lozano-Pérez T. and Wesley M.A. An algorithm for planning collision free paths among polyhedral obstacles. ***Communications of the ACM***, 22(10):560–570, October 1979.

[5] Mazer E. Lozano-Pérez T., Jones J.L and O'Donnell P.A. Task-level planning of pick–and–place robot motions. ***Computer***, 22(3):21–29, March 1989.

[6] Weaver J.M. and Derby S.J. A divide-and-conquer method for planning collision-free paths for cooperating robots. In *AIAA Space Programs and Technologies Conference, Paper AIAA-92-1722*, Huntsville, AL, 1992.

[7] Whitney D.E. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, MMS-10(2):47–53, 1969.

[8] Khatib O. Dynamic control of manipulators in operational space. In *6th IFToMM Congress on Machines and Mechanisms*, New Delhi, 1983.

[9] Vukobratovic M. and Kircanski M. A dynamic approach to nominal trajectory synthesis for redundant manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-14, 1984.

[10] Yoshikawa T. Manipulability of robot mechanisms. *International Journal of Robotics Research*, 4(2):3–9, March 1985.

[11] Hollerbach J. Baillieul J. and Brockett R. Programming and control of kinematically redundant manipulators. In *Proc. 24th IEEE Conf. Decision and Control*, pages 768–774, Las Vegas, NV, December 1984.

[12] Hollerbach J.M. and Suh K.C. Redundancy resolution of manipulators through torque optimization. In *Proc. 1985 IEEE Robotics and Automation Conference*, pages 308–315, St. Louis, MO, March 1985.

[13] Baillieul J. Martin D.P. and Hollerbach J.M. Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Automatic Control*, 5(4):529–533, August 1989.

[14] A. De Luca. Redundant Robots: Introduction to Chapter IX. In M. Spong and F. Lewis, editors, *Robotics*. IEEE Press, 1992.

[15] D.P. Martin, J. Baillieul, and J.M. Hollerbach. Resolution of kinematic redundancy using optimization techniques. *IEEE Transactions on Robotics and Automation*, 5(4), August 1989.

[16] Y. Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison–Wesley Series in Electrical and Computer Engineering: Control Engineering. Addison–Wesley Publishing Company, 1991.

[17] Suh K.C. and Hollerbach J.M. Local versus global torque optimization of redundant manipulators. In *Proc. 1987 IEEE Robotics and Automation Conference*, pages 619–624, Raleigh, NC, March 1987.

[18] Kazerounian K. and Nedungadi A. Redundancy resolution of robotic manipulators at the acceleration level. In *The 7th World Congress on the Theory of Machines and Mechanisms*, pages 1207–1211, Sevilla, Spain, September 1987.

[19] Wang Z. and Kazerounian K. A general formulation for redundancy resolution of serial manipulators. In *Proc. 1990 ASME Design Technical Conference*, pages 373–379, Proc. 1990 ASME Design Technical Conference, September 1990.

[20] P. Jacobs and J. Canny. Robust motion planning for mobile robots. In *1991 IEEE R&A Workshop on Nonholonomic Motion Planning*, Sacramento, CA, April 1991.

[21] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Proc. IEEE 5th Int. Conf. on Advanced Robotics*, pages 1012–1027, Pisa, Italy, June 1991.

[22] P. Jacob, J.-P. Laumond, and M. Taix. A complete iterative motion planner for a car–like robot. *Journees Geometrie Algorithmique*, 1990.

[23] L. Dorst, I. Mandhyan, and K. Trovato. The geometrical representation of path planning problems. Technical report, Philips Laboratories North American Philips Corp., Briarcliff Manor, New York, 1991.

[24] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proc. 1992 IEEE Robotics and Automation Conference*, pages 2533–2540, Nice, France, May 1992.

[25] T. Hague and S. Cameron. Motion planning for nonholonomic industrial robot vehicles. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1275–1280, Osaka, Japan, November 1991.

[26] T. Fraichard and C. Laugier. On line reactive planning for nonholonomic mobile in a dynamic world. In *Proc. 1991 IEEE Robotics and Automation Conference*, pages 432–437, Sacramento, CA, April 1991.

[27] R.W. Brockett. Control theory and singular Riemannian geometry. In *New Directions in Applied Mathematics*, pages 11–27. Springer–Verlag, New York, 1981.

[28] Z.X. Li and J. Canny. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation*, 6(1):62–72, February 1990.

[29] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift: A preliminary report. Technical Report SYCON–90–04, Rutgers Center for Systems and Control, June 1990.

[30] R.M. Murray and S.S. Sastry. Steering nonholonomic systems using sinusoids. In *Proc. 29th IEEE Conference on Decision and Control*, pages 2097–2101, Honolulu, HI, 1990.

[31] Z. Li and L. Gurvits. Theory and applications of nonholonomic motion planning. Technical report, New York University, Courant Institute of Mathematical Sciences, July 1990.

[32] Y. Nakamura and R. Mukherjee. Nonholonomic motion planning of space robots. In *Proc. 1990 IEEE Robotics and Automation Conference*, Cincinnati, OH, 1990.

[33] Z. Vafa and S. Dubowsky. On the dynamics of manipulators in space using the virtual manipulator approach. In *Proc. 1987 IEEE Robotics and Automation Conference*, Raleigh, NC, March 1987.

[34] A.M. Bloch, N.H. McClamroch, and M. Reyhanoglu. Controllability and stabilizability properties of a nonholonomic control system. In *Proc. 29th IEEE Conference on Decision and Control*, pages 1312–1314, Honolulu, HI, 1990.

[35] B. d'Andrea Novel, G. Bastin, and G. Campion. Modelling and control of nonholonomic wheeled mobile robots. In *Proc. 1991 IEEE Robotics and Automation Conference*, pages 1130–1135, Proc. 1991 IEEE Robotics and Automation Conference, April 1991.

[36] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proc. 1990 IEEE Robotics and Automation Conference*, pages 384–389, Cincinnati, OH, May 1990.

[37] C. Samson and K. Ait-Abderrahim. Feedback control of a nonholonomic wheeled cart in cartesian space. In *Proc. 1991 IEEE Robotics and Automation Conference*, Sacramento, CA, April 1991.

[38] C. Samson and K. Ait-Abderrahim. Feedback stabilization of a nonholonomic wheeled mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1242–1247, Osaka, Japan, November 1991.

[39] H.J. Sussmann and Y. Chitour. A continuation method for nonholonomic path finding problem. IMA Workshop on Robotics, January 1993.

[40] A. Divelbiss and J.T. Wen. A perturbation refinement method for nonholonomic motion planning. In *Proc. 1992 American Control Conference*, Chicago, IL, June 1992.

[41] A. Divelbiss and J.T. Wen. Nonholonomic motion planning with constraint handling: Application to multiple–trailer vehicles. In *Proc. 31th IEEE Conference on Decision and Control*, Tucson, AZ, December 1992.

[42] S. Seereeram and J.T. Wen. A global approach to path planning for redundant manipulators. Accepted for publication in IEEE Trans. on Robotics and Automation, 1993.

[43] E.D. Sontag and Y. Lin. Gradient techniques for systems with no drift. In *Proceedings of Conference in Signals and Systems*, 1992.

[44] E.D. Sontag. Non–singular trajectories, path planning, and time–varying feedback for analytic systems without drift. IMA Workshop on Robotics, January 1993.

[45] Dorny C.N. *A Vector Space Approach to Models and Optimization.* R. E. Krieger Publishing Co., Florida, 1986.

[46] D.G. Luenberger. *Linear and Nonlinear Programming.* Addison–Wesley, 2 edition, 1984.

[47] G. Oriolo and Y. Nakamura. Free–joint manipulators: Motion control under second–order nonholonomic constraints. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Osaka, Japan, November 1991.

[48] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge, U.K., 1986.

[49] A. Divelbiss and J.T. Wen. A perturbation refinement method for nonholonomic motion planning. CAT report #10, Rensselaer Polytechnic Institute, March 1992.

[50] J.F. III Watson, D.R. Lefebvre, S.H. Murphy A.A. Desrochers, and K.R. Fieldhouse. Testbed for cooperative robotic manipulators. In A.A. Desrochers, editor, *Intelligent Robotic Systems for Space Exploration.* Kluwer Academic Publishers, 1992.